

2.5 Контрольні запитання і завдання

1. Що таке статичне та динамічне прогнозування. Наведіть приклади.
2. Чому не періодичні переходи приводять до максимальних витрат продуктивності процесору?
3. Проаналізуйте код, який формується при реалізації циклів типу for, while, do, сформууйте рекомендації по їх застосуванню.
4. Проаналізуйте код, який формується за допомогою оператора switch, та час виконання коду з використанням switch та заміною його еквівалентними операторами if. Сформууйте рекомендації по їх застосуванню.

3 ВИКОРИСТАННЯ SIMD КОМАНД ПРИ РОЗРОБЦІ ПРОГРАМ

3.1 Мета роботи

Навчитися ефективно використовувати SIMD команди сучасних процесорів при складанні програм

3.2 Підготовка до роботи

1. Вивчити класифікацію обчислювальних систем згідно Flynn. Вивчити типи SIMD команд на сучасних процесорах (лекція 1, 2).
2. Вивчити методи визначення системної інформації про процесори, яка необхідна для визначення можливості використання різних типів SIMD команд (таблиця 3.1).

Таблиця 3.1 Параметри функції `__cruid` для отримання інформації про процесор

Функція (b)	Результатр
0	а [0] – максимальный номер звичайної функции; а [1], а [3], а [2] – тип процессора. Для Intel повинні отримати рядок “GenuineIntel” – оригінальний Intel, Для AMD повинні отримати рядок “AuthenticAMD”.
1	а [0] – додаткова інформація про тип процесора (тип, родина,

	модель,...); а [1] – Біти 8-15 – розмір рядка кешу в 8 – байтових елементах; Біти 16-23 – максимальна кількість ядер процесору; а [2] – Біт 0 – підтримка SSE3; а [2] – Біт 9 – підтримка SSSE3; а [2] – Біт 19 – підтримка SSE4.1; а [2] – Біт 20 – підтримка SSE4.2; а [2] – Біт 26 – підтримка AVX; а [2] – Біт 27 – підтримка AVX; а [2] – Біт 28 – підтримка AVX; а [3] – Біт 23 – підтримка MMX; а [3] – Біт 25 – підтримка SSE; а [3] – Біт 26 – підтримка SSE2; а [3] – Біт 28 – підтримка HYPER Threading;
0x80000000	а [3] – Біт 31 – підтримка 3DNow! (AMD); а [3] – Біт 30 – підтримка 3DNow!2 (AMD);
0x80000001	а [2] – Біт 6 – підтримка SSE4A; а [2] – Біт 11 – підтримка SSE5;
0x80000002 - 0x80000004	а [0], а [1], а [2], а [3] – повна назва процесора та його тактова частота (INTEL, AMD).

3. Вивчити призначення типів даних `__m128`, `__m256`, `__m128d`, `__m128i`.

Дані довжиною 128 біт оброблюються тільки SSE командами.

`__m128` – дані з плаваючою крапкою звичайної точності;

`__m128d` – дані з плаваючою крапкою подвійної точності;

`__m128i` – цілі дані. Компоненти довжиною 8, 16, 32, 64 біта.

Дані довжиною 256 біт оброблюються тільки AVX командами.

4. Навчитися використовувати SSE та AVX функції з використанням мови високого рівня (заголовний файл `intrin.h` і файли, які він підключає).

3.3 Порядок виконання лабораторної роботи

1. Скласти функції для визначення типу процесору, а також можливості використання SSE та AVX (AVX2) команд різного типу.

2. Скласти функції для обчислення: $x[i] = |y[i]| + |z[i]|$ ($i = 0.. n-1$) для даних наступних типів без використання SIMD команд з найбільшою довжиною блоку, яка підтримується : (суми чисел за модулем)

- цілі довжиною 8 біт;
- цілі довжиною 16 біт;
- цілі довжиною 32 біт;
- цілі довжиною 64 біт;
- з плаваючою крапкою звичайної точності;
- з плаваючою крапкою подвійної точності.

3. Порівняти обчислювальну складність для цих функцій для довжини масиву $4096 * 4096$.

4. Результати виконання пунктів 2, 3 занести в таблицю.

5. Зробити висновки по ефективності використання SIMD команд для виконання арифметичних операцій.

6. Скласти функції для обчислення $x[i] = \sqrt{y[i]}$ ($i = 0.. n-1$) для даних з плаваючою крапкою звичайної та подвійної точності без використання та з використанням SIMD команд.

7. Порівняти обчислювальну складність для цих функцій для довжини масиву 4096.

8. Скласти функцію для обчислення $x[i] = y[i] * z[i]$ з комплексними даними, компоненти яких дані типу float без та з використанням SIMD команд. Зробити висновки по ефективності.

9. (Вищий балл) Дослідити ефективність команд зрушення (SSE) для зрушення даних довжиною 512 біт на 1 біт.

3.4 Зміст звіту

1. Код програми з функціями відповідно завданню.
2. Таблиця зі значеннями обчислювальної складності для усіх функцій.
3. Висновки по ефективності використання SIMD команд для виконання арифметичних операцій і обчислення $\sqrt{}$.

3.5 Контрольні запитання і завдання

1. Які типи обчислювальних систем згідно Flynn ви знаєте? До який типів відносяться сучасні процесори?
2. Які недоліки мають MMX команди?

3. Які недоліки мають 3DNow! команди?
4. Які недоліки мають SSE команди?
5. Вам треба обробляти дані типу __m256i. Яку перевірку необхідно обов'язково виконати перед використанням команд для обробки даних цих типів?
6. Як виділити пам'ять під масив довільної довжини таким чином, щоб можна було для цього масиву використовувати AVX команди?

4 ПОРІВНЯННЯ МЕТОДІВ СТВОРЕННЯ ПОТОКІВ В WINDOWS I OPEN MP

4.1 Мета роботи

При запуску програми операційна система створює один потік. Якщо виконується одна програма, то фактично використовується одне ядро процесору. Для забезпечення завантаження ядер необхідно, щоб одночасно необхідно було виконувати декілька потоків. Краще, якщо кількість готових потоків в кожен момент часу дорівнював числу ядер. Таким чином, проблема створення потоків, керуванням їх кількістю в залежності від кількості ядер, є актуальною задачею. Метою даної лабораторної роботи є порівняння методів створення потоків за допомогою потоків OS Windows і Open MP з боку витрат часу програміста на створення і додаткових витрат часу процесору.

4.2 Підготовка до роботи

4.2.1 Включення режиму Open MP

Створити проект⁴.

В закладці Project-><Ім'я проекту>Properties->Command Line в окні Additional Options додати ключ трансляції /openmp або в Project->< Ім'я проекту>Properties->Language встановити флажок OpenMP Support і натиснути кнопку Примінити, а потім ОК.

Перевірити, що режим використання OpenMP вмикнувся. При включенні цього режиму визначається змінна _OPENMP. Для перевірки наявності цієї змінної можна використовувати наступний код::

```
#include <stdio.h>
#include <omp.h>

int main() {
    #ifdef _OPENMP
        printf ("_OPENMP Defined\n");
    
```

⁴ Деякі версії компіляторів не працюють в разі створення Solution з декількома проектами